REMARKS

The present response accompanies a Request for Continued Examination (RCE). Claims 1-5, 7-12, 14-17, 19-22, 24, and 30-35 are currently pending in the present application. Support for the amendments may be found throughout the specification, such as paragraphs [0045] – [0055], for example. No new matter has been added by way of the amendments.

Interview Summary

Applicant's representatives, Mr. Kenneth Eiferman and Mr. Bentley Olive, and Examiner Robert Timblin participated in a telephonic interview on July 6, 2009, to discuss the claim amendments herein. Examiner Timblin agreed to reevaluate the rejections in view of the amendments and remarks herein.

Claim Objections

Claim 24 stands objected to under the contention that it is unclear where the platform name is passed to in the phrase "passing the storage platform path name" (Office Action dated May 29, 2009 ("Office Action") at p. 2). The Office Action states that, in correspondence to claim 1, the phrase can be interpreted to mean that the storage platform path name is passed to a database server (*Id.*). Claim 24 has amended the phrase as follows: "passing the storage platform path name to the storage platform". Applicants respectfully submit that this amendment sufficiently clarifies that the storage platform path name is being passed to a database server.

The Office Action also states that the features lead by "identifying" and "passing" in claim 24 should instead begin with "identify" and "pass" (*Id.*). The features have been amended as suggested by the Office Action. For the same reasons, the term "performing" in claim 24 has been replaced with the term "perform".

Accordingly, for the above reasons, it is respectfully submitted that the objection to claim 24 should be withdrawn.

Claim Rejections – 35 U.S.C. § 103

Claims 1-9 and 15-20 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,922,708 to Sedlar ("Sedlar") in view of U.S. Patent No. 7,035,874 to Reed et al. ("Reed"). Without conceding the merits of the rejection, Applicants have amended independent claims 1, 14, 19, and 24 to further clarify the claimed subject matter.

Claim 1 has been amended to recite, in part, a method for executing a transaction comprising at least one query language statement and at least one file system statement. Each statement relates to a user defined type ("UDT") associated with a database server. Further, claim 1 recites storing at least one field relating to the UDT in a relational database table. At least one field is a filestream field. Data for each filestream field is stored in a respective file separate from the relational database table. The method also comprises: receiving each of the at least one file system statements. Each statement comprises a call to open a first item and at least one of a call to read from the first item and to write to the first item, and a call to close the first item; and receiving each of the at least one query language statement, wherein each query language statement is associated with a second item. For each file system statement, the method includes, upon detecting a storage platform path name associated with the first item in a file system statement: forwarding the file system statement to an agent, wherein the agent performs a call to the storage platform by passing the storage platform path name to the storage platform.

Further, the claimed method comprises identifying the first item based upon the storage platform path name; passing a database engine function that returns a file system path name corresponding to the first item to the database server; performing a table look-up for the UDT associated with the first item on the database server; extracting a real file system path corresponding to the first item using the database engine function; using the real file system path to perform an operation on the first item by passing the real file system path back to the agent, wherein the agent then interacts with the file system to cause execution of file system statement. If the file system statement includes open, read and close operations, a transaction is created as part of an open operation. ***The transaction is managed separately from the database server and comprises the at least one query language statement and the at least one file system statement.*** Further, if the file system statement includes open, read and close operations, the method implements the following steps: ***determining whether the at least one query language statement conflicts or the at least one file system statement***

conflicts with another transaction; and ***resolving the conflict if the at least one query language statement conflicts or the at least one file system statement conflicts with another transaction***. ***If the at least one query language statement and the at least one file system statement do not conflict with another transaction***, the method implements the following steps: obtaining a read lock on a data table row associated with the associated first item for the file system statement; performing a read operation in the context of the transaction; and committing the transaction as part of a close operation.

The claim also recites that if the file system statement includes open, write and close operations, the method includes creating a transaction as part of an open operation. ***The transaction is managed separately from the database server and comprises the at least one query language statement and the at least one file system statement***. Further, if the file system statement includes open, write and close operations, the method implements the following steps: ***determining whether the at least one query language statement conflicts or the at least one file system statement conflicts with another transaction***; and resolving the conflict if the at least one query language statement or the at least one file system statement conflicts with another transaction. ***If the at least one query language statement conflicts and the at least one file system statement do not conflict with another transaction***, the method implements the following steps: obtaining a write lock on a data table row associated with the associated first item for the file system statement; performing a write operation in the context of the transaction; committing the transaction as part of a close operation; and, for each query language statement, starting a transaction on the database server updating fields associated with the second item in the query language statement and sending an updategram to the database server.

Therefore, in summary, claim 1 has been amended to recite (1) ***determining whether a query language statement or a file system statement of the same transaction conflicts with another transaction***; (2) ***resolving the conflict if the query language statement or the file system statement conflicts with another transaction***; and (3) ***obtaining a write or read lock on a data table row associated with an associated first item for the file system statement if the query language statement and the file system statement do not conflict with another transaction***.

For example, referring to Fig. 3 of the present application, a storage platform 308 receives statements from a client application 300 to be executed upon items stored in a data store 310 (act 410) (Present Application at ¶ [0045]). Such statements can be query language statements or file system statements (*Id.*). A query language statement may include, for example, either a SELECT statement or an UPDATE statement (*id.* at ¶ [0046]). A file system statement may include an open operation, either a read operation or a write operation, and a close operation (*id.* at ¶ [0047]). The statements can be a part of a transaction (*id.* at ¶¶ [0046] and [0047]).

At act 414, it is determined whether starting the transaction will result in a conflict (*id.* at ¶ [0049]). For example, if any statements within the transaction correspond to a row upon which another transaction has already acquired a write lock, then a conflict will occur (*Id.*). If it is determined that a conflict will occur, then the conflict is resolved (act 416) (*id.* at ¶ [0050]). At act 418, the transaction is started by acquiring read locks and write locks on the appropriate rows (*id.* at ¶ [0051]).

Sedlar does not disclose or suggest the claim 1 features of (1) ***determining whether a query language statement or a file system statement of the same transaction conflicts with another transaction***; (2) ***resolving the conflict if the query language statement or the file system statement conflicts with another transaction***; and (3) ***obtaining a write or read lock on a data table row associated with an associated first item for the file system statement if the query language statement and the file system statement do not conflict with another transaction***. Sedlar relates to a method for performing file system statements. Generally, Sedlar teaches that calls to perform file system operations are received through a file system interface. The file system operations are performed as a single transaction by performing the steps of: if all file system operations of the plurality of file system operations are completed without failure, making permanent all changes made by the plurality of operations and if any of the file systems operations fail, then undoing all changes made by all the file operations.

Further, Sedlar teaches hierarchical indexes that support a pathname-based access method of a hierarchical system, moving from parent items to their children, as specified by the pathname (Sedlar at col. 7, lines 33-36). A files table 710 shown in FIG. 7 is used to store the files illustrated in FIG. 6 within a relational database (*id.* at col. 7, lines 43-45). Sedlar also teaches that searching an index for a target file stored in the files table 710 (*id.* at col. 10, lines 8-24). The hierarchical index can be used to locate an exemplary Example.doc file, which may be stored in the files table 710 (*Id.*). However, Sedlar does not teach or suggest the aforementioned claim 1 features.

Reed also does not teach or suggest the claim 1 features of (1) ***determining whether a query language statement or a file system statement of the same transaction conflicts with another transaction***; (2) ***resolving the conflict if the query language statement or the file system statement conflicts with another transaction***; and (3) ***obtaining a write or read lock on a data table row associated with an associated first item for the file system statement if the query language statement and the file system statement do not conflict with another transaction***. Rather, Reed relates to methods and apparatuses for streaming data from a database system to a client system. More particularly, Reed teaches a MediaUDT object for representing a MediaUDT field (Reed at col. 4, lines 19 and 20). There is no disclosure or suggestion anywhere in Reed of the aforementioned claim 1 features.

For at least the reasons set forth above, Sedlar and Reed, either alone or in combination, do not teach or suggest the aforementioned claim 1 features. Claim 6 has been cancelled. Accordingly, it is respectfully submitted that the rejection of claim 1 and its dependent claims 2-12 under 35 U.S.C. § 103(a) should be withdrawn.

Independent claims 14, 19, and 24 have been amended similar to claim 1. Therefore, for at least the reasons set forth above with respect to claim 1, Applicants respectfully submit that Sedlar and Reed, either alone or in combination, do not teach or suggest the features of each of claims 14, 19, and 24. Claims 25-29 have been cancelled. Accordingly, it is respectfully submitted that the rejection of claims 14, 19, and 24 and their dependent claims 15-18, 20-23, and 30-35 under 35 U.S.C. § 103(a) should be withdrawn.

CONCLUSION

In view of the foregoing, Applicants respectfully submit that the claims are allowable and that the present application is in condition for allowance. Reconsideration of the application and an early Notice of Allowance are respectfully requested. In the event that the Examiner cannot allow the present application for any reason, the Examiner is encouraged to contact the undersigned attorney, Kenneth R. Eiferman, to discuss the resolution of any remaining issues.

Date: August 6, 2009

/Kenneth R. Eiferman/
Kenneth R. Eiferman
Registration No. 51,647

Woodcock Washburn LLP
Cira Centre
2929 Arch Street, 12th Floor
Philadelphia, PA 19104-2891
Telephone: (215) 568-3100
Facsimile: (215) 568-3439